Project Documentation

Class: ISYS 567 – Internship

Instructor: Prof. Verma

Students: Brandon Lai
Pascal Schuele

Table of Contents

1.) Introduction to Cloud Computing

Cloud computing is the new frontier for technology, with the ability to scale from home use to enterprise use in the blink of an eye. Cloud computing, at its simplest, is just services that are being delivered to the user by use of a network (usually the Internet). There are two main services that cloud services provide, compute and storage. With the compute cloud comes a flurry of other services, such as Software as a Service (SaaS), Platform as a Service (PaaS), Desktop as a Service (DaaS), Security as a Service (SECaaS) and more. The storage service of cloud just acts like folder, which just holds the user's data. This provides a centralized system for the files. These services are versatile and most people feel it is more secure to run them on-site. In the cloud computing business, there are two main types of clouds, public and private.

In our project we will explain the main differences between these different types of clouds. We will take a quick look through the different options for private cloud software. After that, we will single out one up and coming open source software (Openstack) to implement a private cloud. Finally we provide a How-To with all the instructions needed, to set up a private cloud with Openstack.

2.) Public vs. Private Cloud

Public clouds are cloud systems that are available for everyone's use, such as Dropbox and Amazon's EC2. These services can be both free and subscription based, depending on the user's needs. A private cloud can offer the same services as a public cloud, however, its services are limited to people behind the company's firewall. There are a few reasons to choose a private cloud over a public cloud, and these options will be explored here. There are a couple main reasons why cloud computing has been on the rise. One reason is elasticity and scalability. "This encompasses the idea of computing on demand, and the ability to increase the supply of computing resources as they are needed to deal with spikes in demand for a particular application or service. There's also the idea of turning computing resources into a commodity so more can be

added over time, as needed, to ensure systems are almost infinitely scalable."[1] This allows variable amounts of users on at the same time, or even enabled whenever it is necessary to alleviate bottlenecks.

There is a downside to private clouds, cost. The capital of hosting a private cloud is substantially more than having a public cloud hosted. The need to purchase the hardware, as well as hiring an administrator to manage the private cloud is the main downsides of it.

In general, public clouds are highly structured and automated. Usually it's not possible for enterprises to get particular SLA's (Service Level Agreement) for their specific needs. Public clouds usually come with a standard SLA, so you just have the choice to take it or leave it. This is one important fact, any organization which thinks about using public clouds, should always keep in mind: You have to adapt to the cloud, the cloud doesn't adapt to you. Basically you should first think about how critical the data is which you want to store in a cloud. The more critical the data is, the more important is it to keep them in house.

3.) Open source software for private clouds

Open source software for private clouds is widely available. Openstack, Eucalyptus, Ganeti, OpenNebula are the main competitors in the private cloud area. All of these except for Ganeti provide a clean user GUI, so we will just compare Openstack, Eucalyptus and OpenNebula. Out of the 3 remaining, OpenNebula focuses tremendously on Data Center virtualization, and has fewer options for controlling what we want from a private cloud. Therefore, we are left with Eucalyptus and Openstack. With Eucalyptus, there are many tools that are comparable to Openstack's, however, Openstack has a more user-friendly GUI and the functions seem simpler to use and deploy. After taking a look at the available software, Openstack fits our needs best.

[1] P. Rubens, January 27, 2010 - http://www.serverwatch.com/trends/article.php/3861191/Private-Cloud-Defined.htm

4.) Openstack as example of open source software for private clouds

"Openstack is a cloud operating system that controls large pools of compute, storage, and networking resources throughout a datacenter, all managed through a dashboard that gives administrators control while empowering their users to provision resources through a web interface."[2] Openstack has many features that make deploying and managing a private cloud simple. "Openstack is architected to provide flexibility as you design your cloud, with no proprietary hardware or software requirements and the ability to integrate with legacy systems and third party technologies. It is designed to manage and automate pools of compute resources and can work with widely available virtualization technologies, as well as bare metal and high-performance computing (HPC) configurations"[3]. We are able to run Openstack on top of Ubuntu. Openstack has the capabilities to make the transition from other cloud services with load balancing, which allows it to migrate the services without downtime to the end users. Since Openstack has met the criteria for this project, we will move forward and install it on our test server running Ubuntu 10.4.

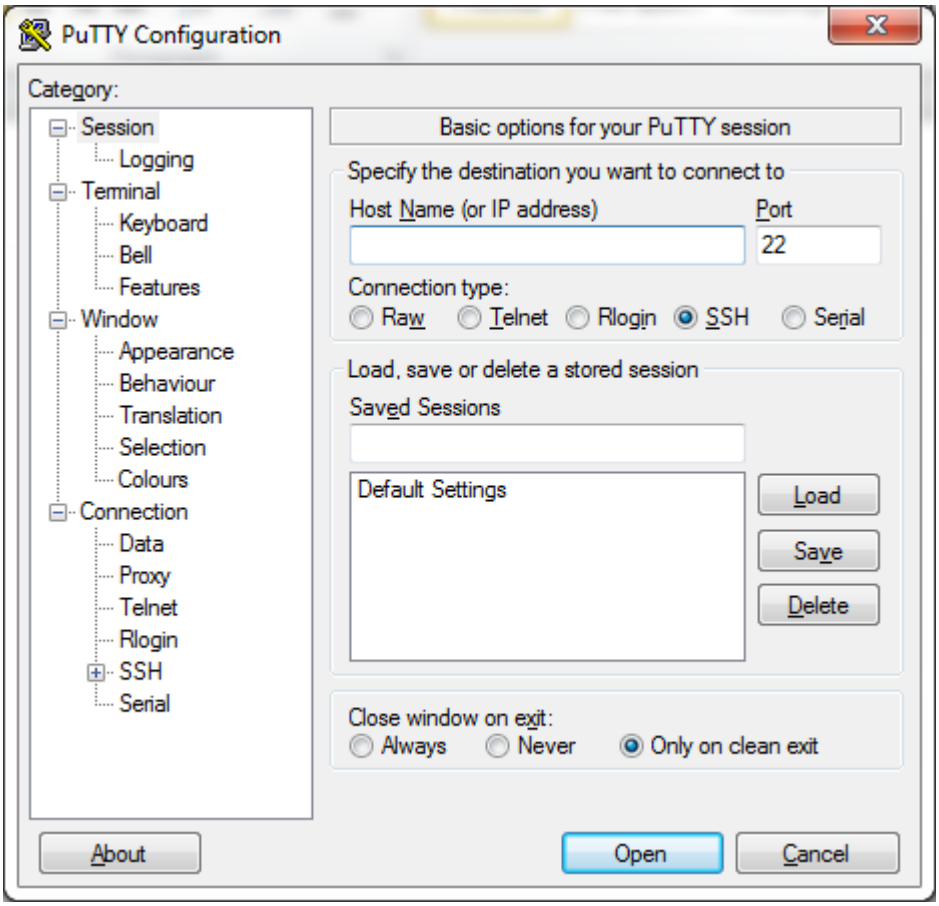
[2] http://www.Openstack.org/software/

[3] http://www.Openstack.org/software/Openstack-compute/

5.) How-To set up a private cloud using Openstack

5.1) SSH connection to the server

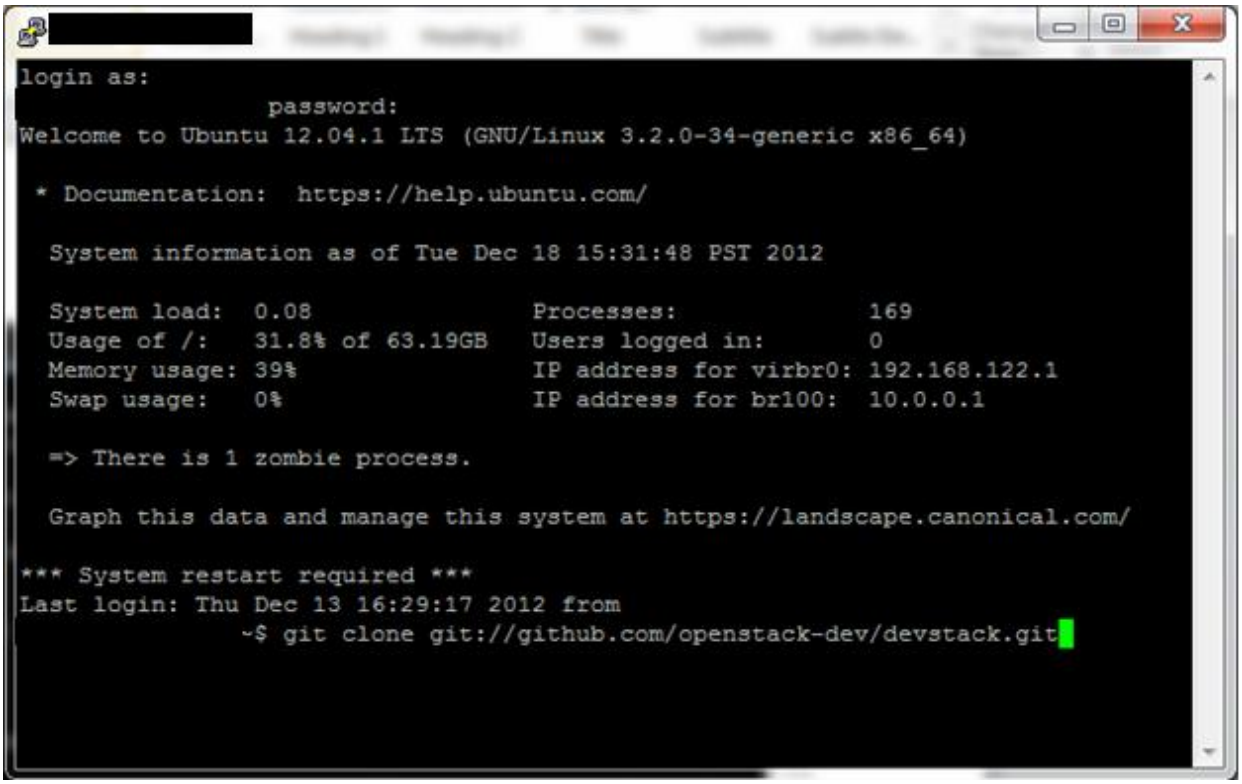
First, we have to connect to the server via a Secure Shell (SSH). Therefore we are using the assigned hostname “bus-mac16” with the standard TCP port 22. One of the free tools is PuTTY, which is an open source SSH and telnet client. The latest version can be downloaded at: http://www.putty.org



5.2) Downloading devstack

After the connection is established, we login on the shell and download devstack to the server using the following command:

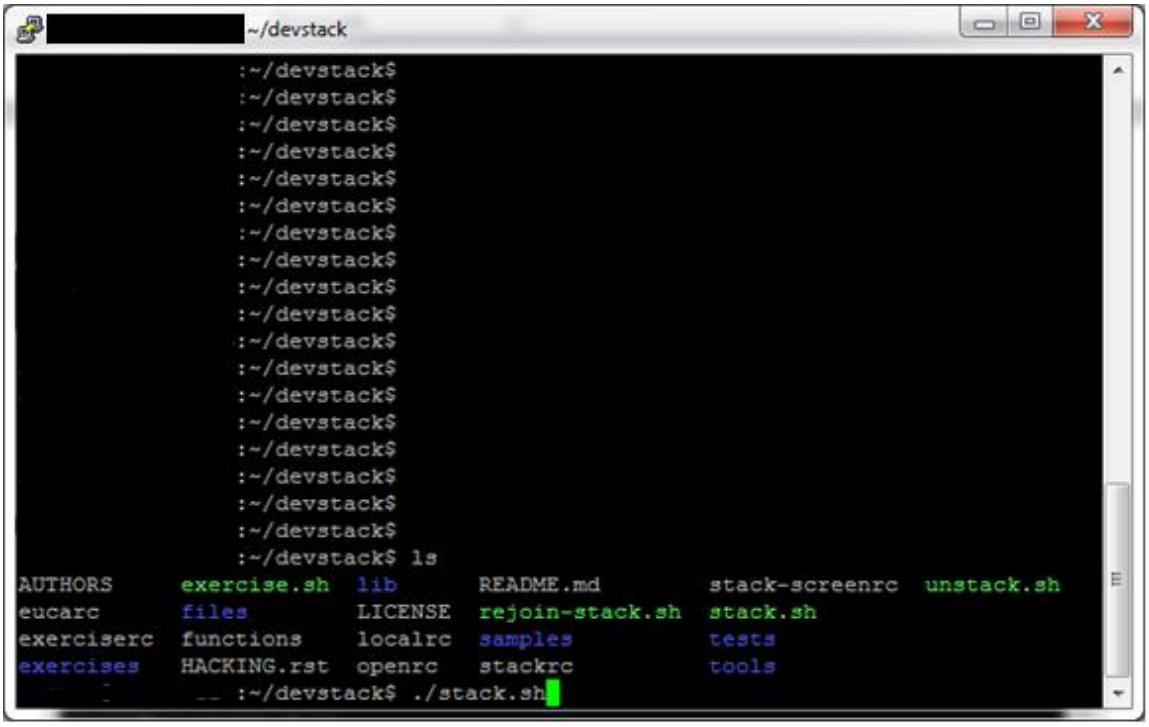
"git clone https://github.com/Openstack-dev/devstack.git"[4]



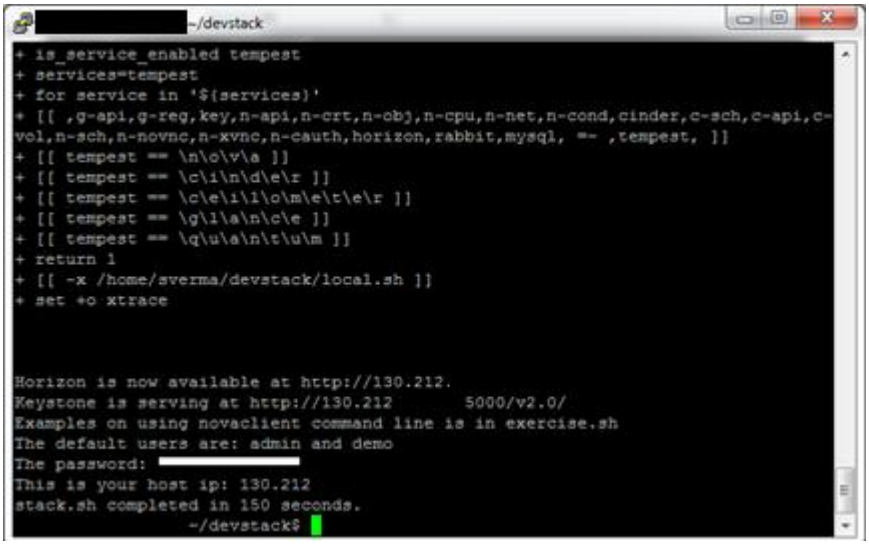
[4] http://devstack.org/

5.3) Execution of the script “stack.sh”

Afterwards we switch to the “/devstack” directory and execute the “stack.sh” script by using the “screen” program “./”

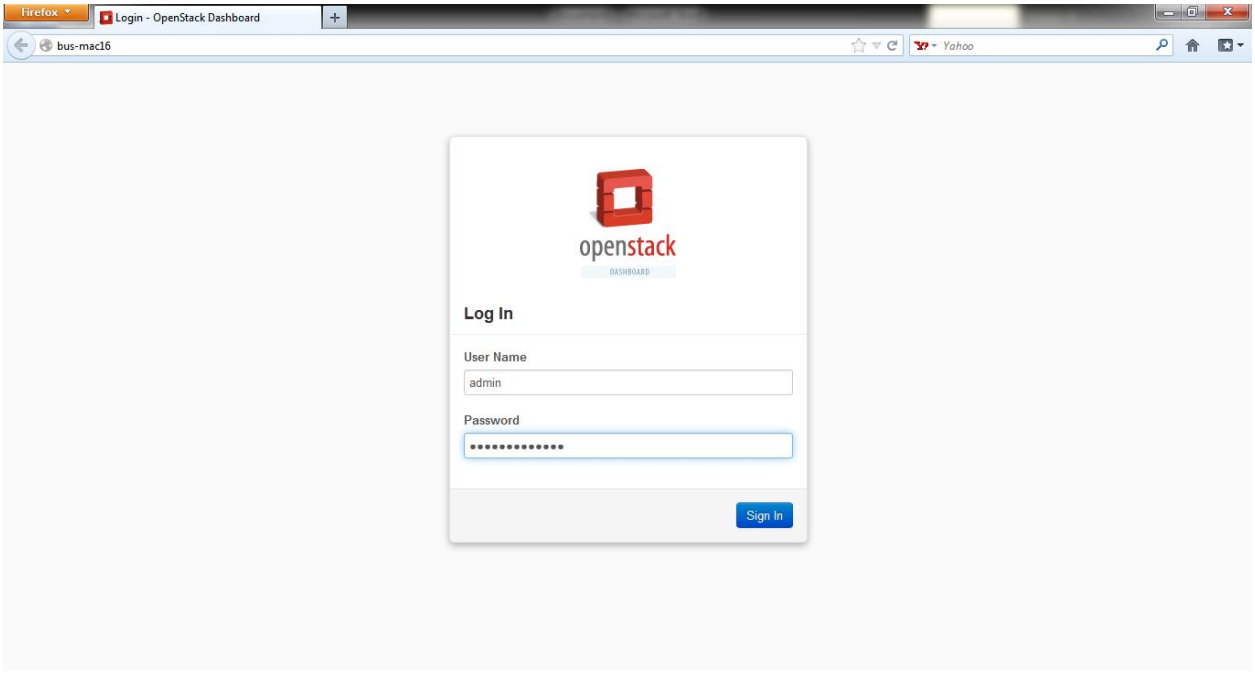


After the script was successfully executed, it shows the configuration and under which IP the system is reachable.



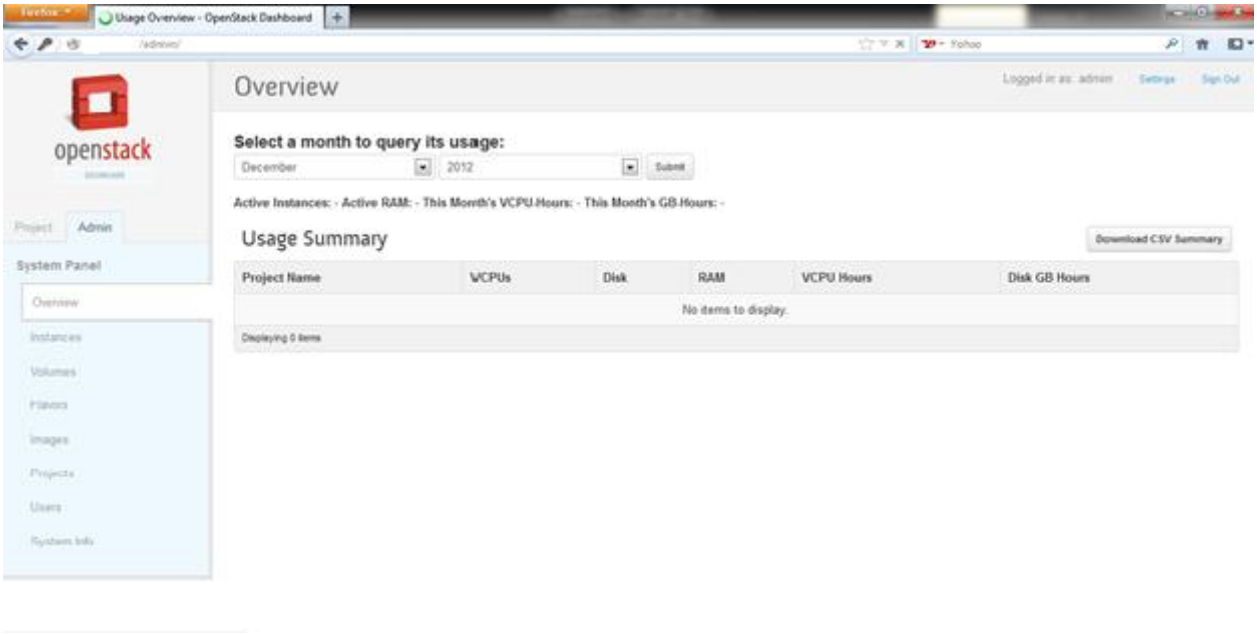
5.4) Log In to Openstack

From now on, the system is also available under the GUI with using any web browser. It's possible to either connect to the IP address or the DNS name. As you can see in the following screenshot we used the DNS name because it's easier to remember. Horizon manages the GUI for Openstack.



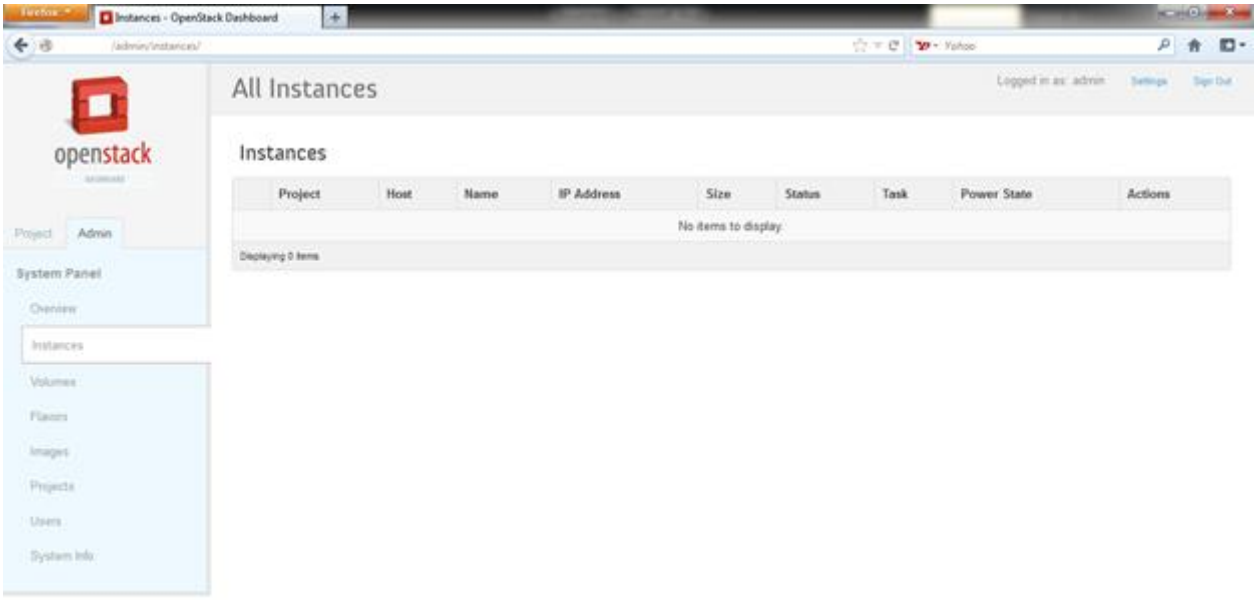
5.5) Overview

The Overview page is the default page after you log into Openstack. There are many options on the left hand side to choose from, the Overview screen allows users to see which projects are currently running, along with details about the projects.



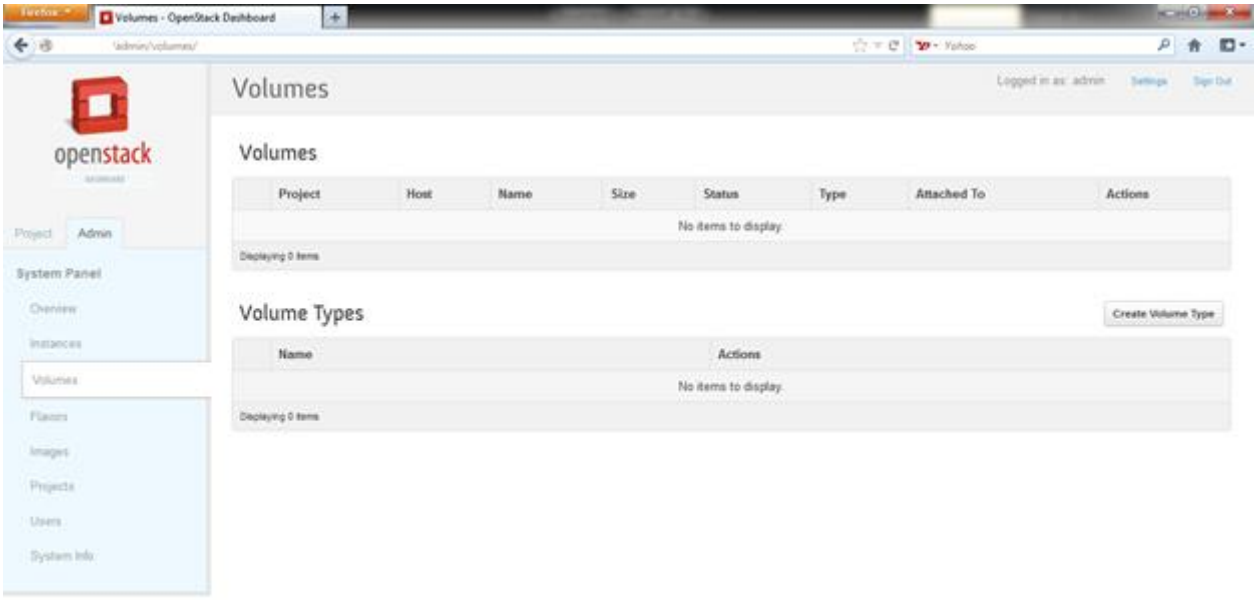
5.6) Instances

The Instances page is next, it shows which instances are running, as well as the IP address, Size, Status, Task, Power State and management options for each instance. Instances are individual VM's that are running on physical compute nodes. Nova handles the instances for Openstack.



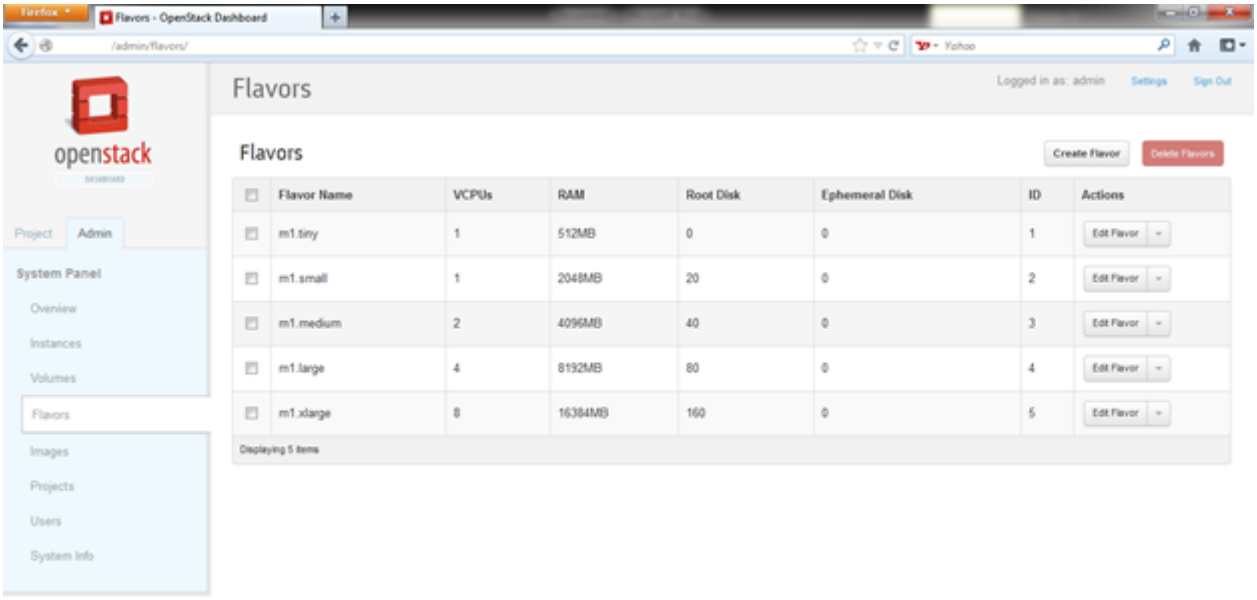
5.7) Volumes

In this installation, Cinder is installed instead of Swift. Cinder is for block storage. On the Volumes page, Openstack shows which volumes are active at the moment. We are able to create different volume types with the Create Volume Type button.



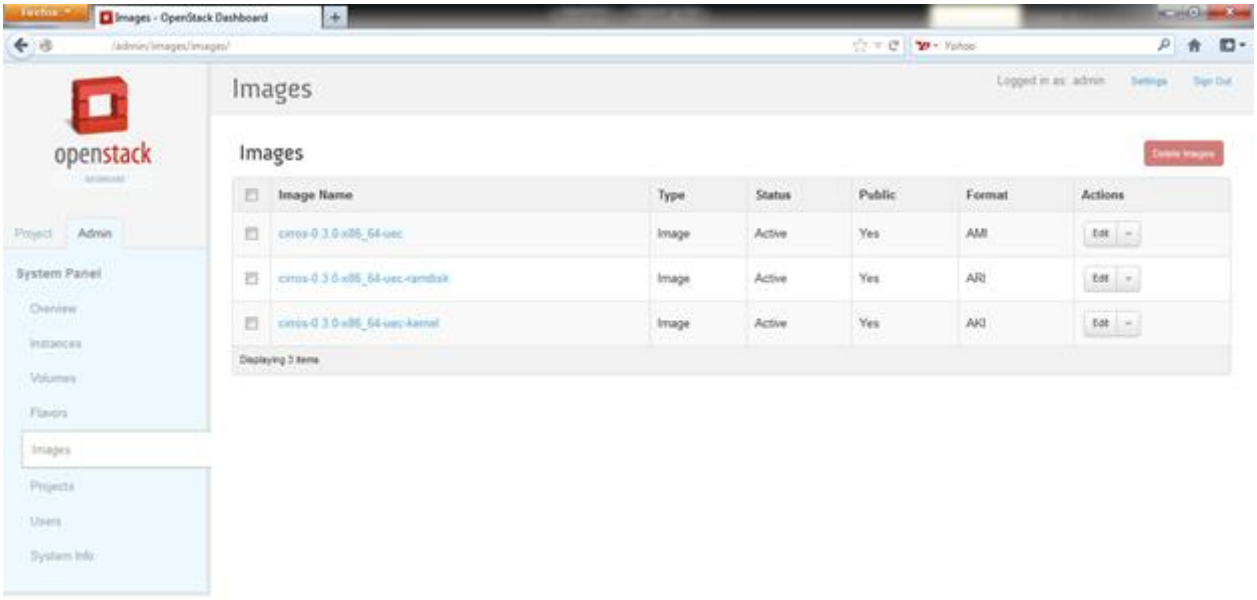
5.8) Flavors

Flavors is the next page. Flavors are customizable settings to create each virtual machine with. The created machines with have the specs of the flavor. The configureable options for flavors are VCPUs, RAM, Root Disk, and Ephemeral Disk. Ephemeral disks are local disk devices. This makes it easy to customize each VM to meet performance needs.



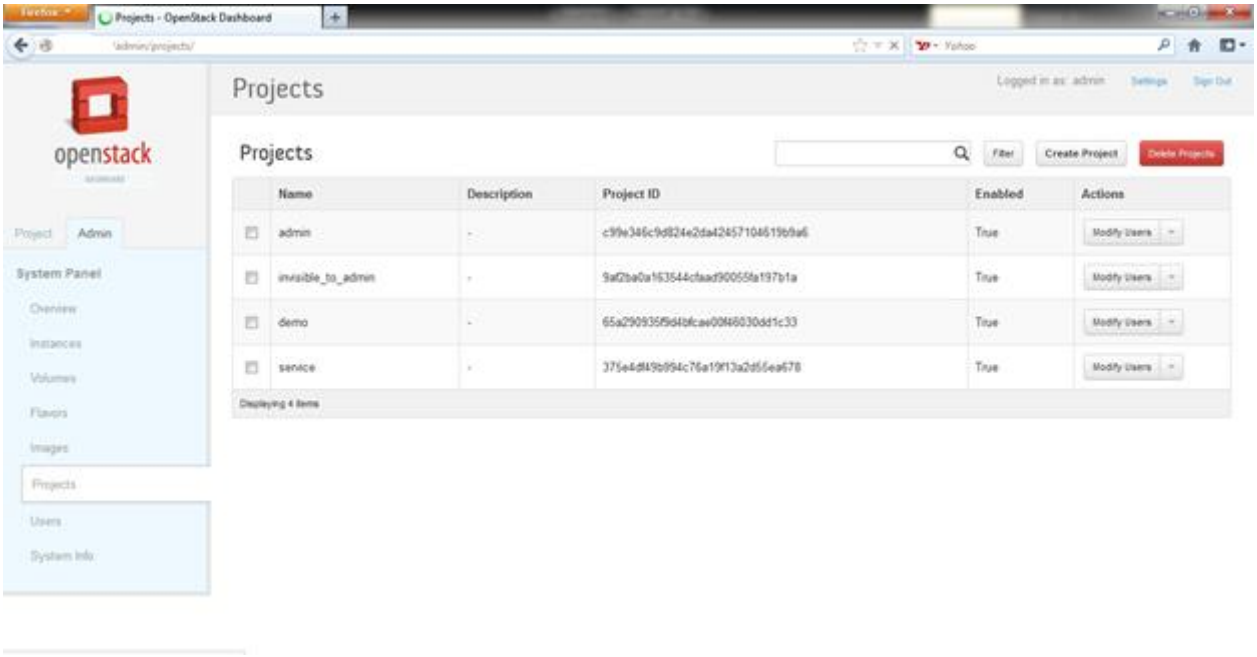
5.9) Images

Images is the next page. This is where users can choose specific images for Openstack to deploy. The default image is cirros 0.3.0 x86_64. Images can be easily added by editing the stackrc text tile in the command line. Glance is responsible for storage and management of the images.



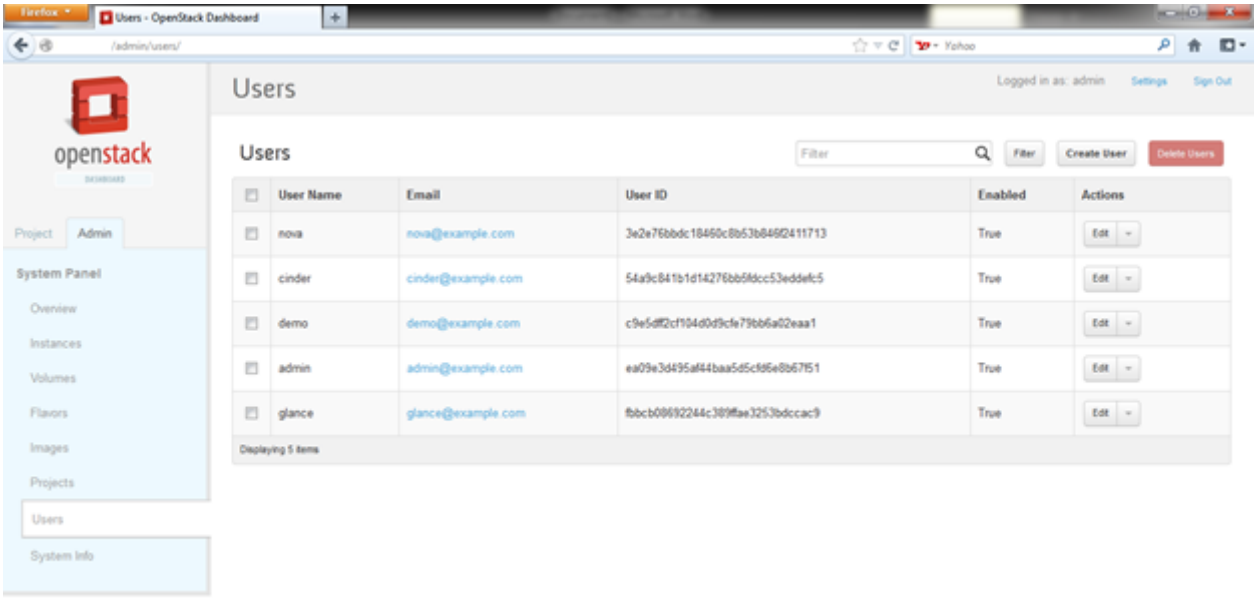
5.10) Projects

Projects are the next page. These and Users are basically the same thing, they were just different words for the different versions of Openstack. These are basically the user accounts. Keystone handles the managment of users and projects.



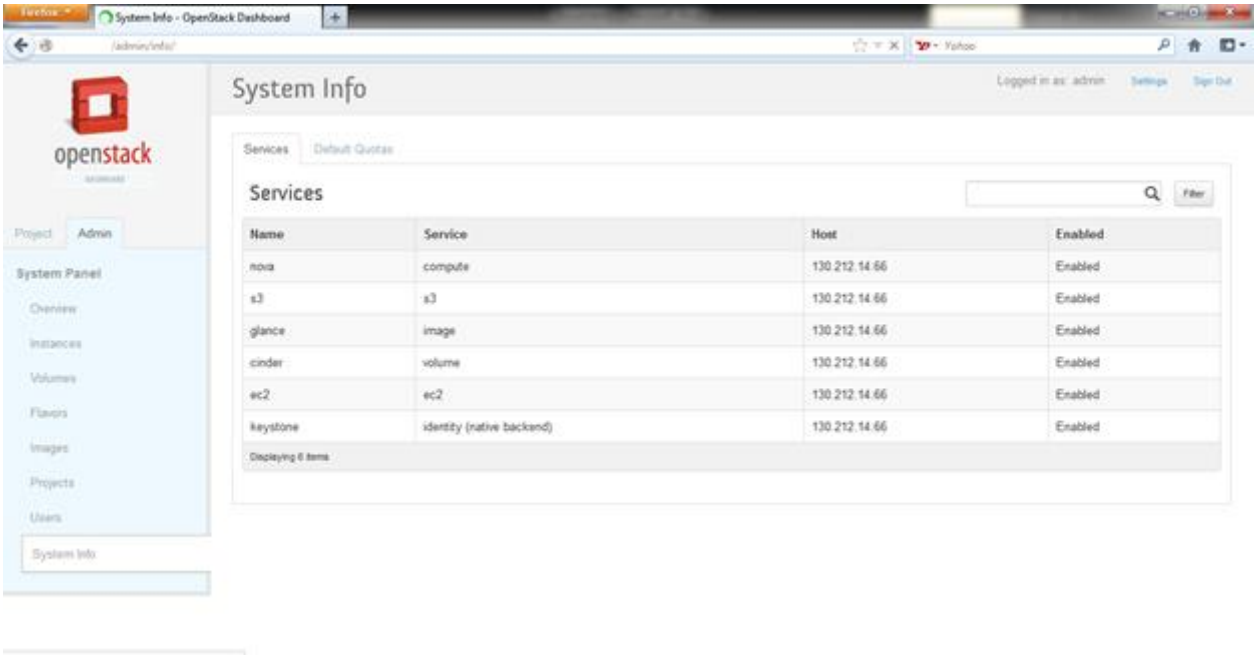
5.11) Users

Users are the next page. These and Projects are basically the same thing, they were just different words for the different versions of Openstack. These are basically the user accounts. Keystone handles the managment of users and projects.



5.12) System Info

System Info is next, which just depicts which services are running, what the name of the services are, the host IP address and if they are enabled or disabled.

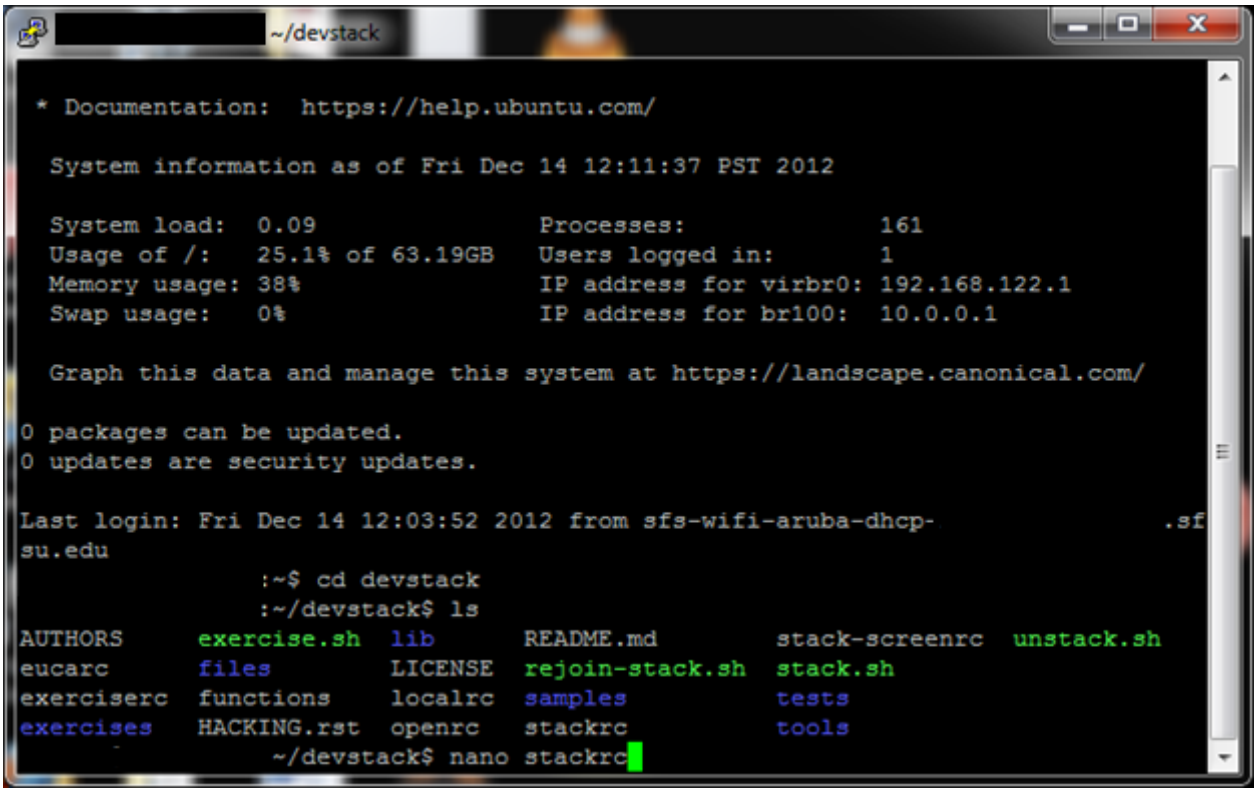


5.13) Adding an image to Openstack

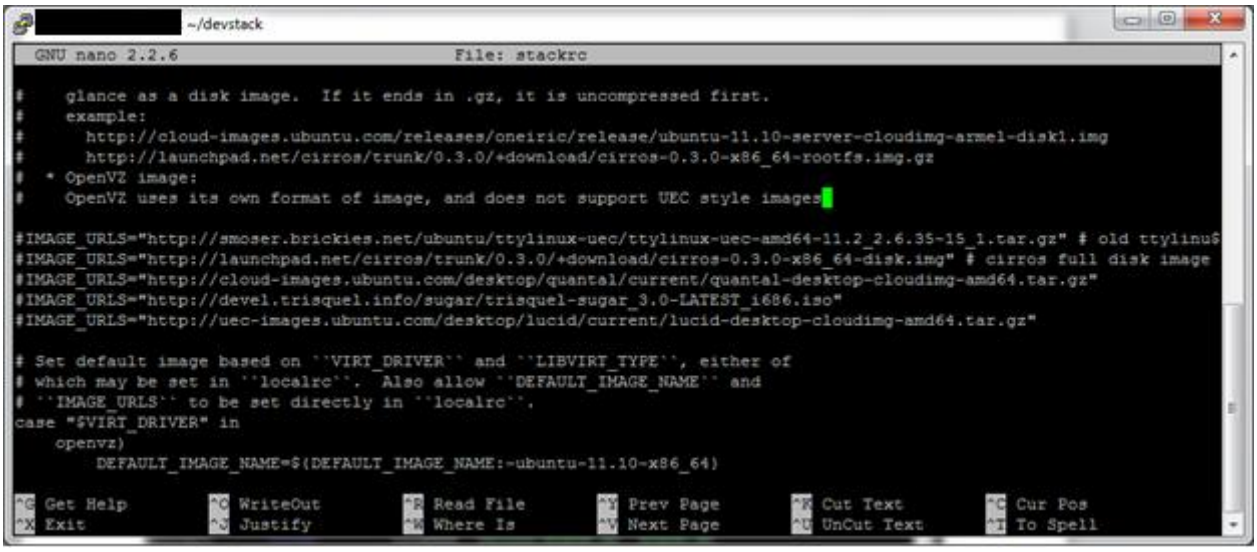
Open up the /devstack directory.

Use the command open up *stackrc* using the text editor *nano*.

"Nano stackrc"

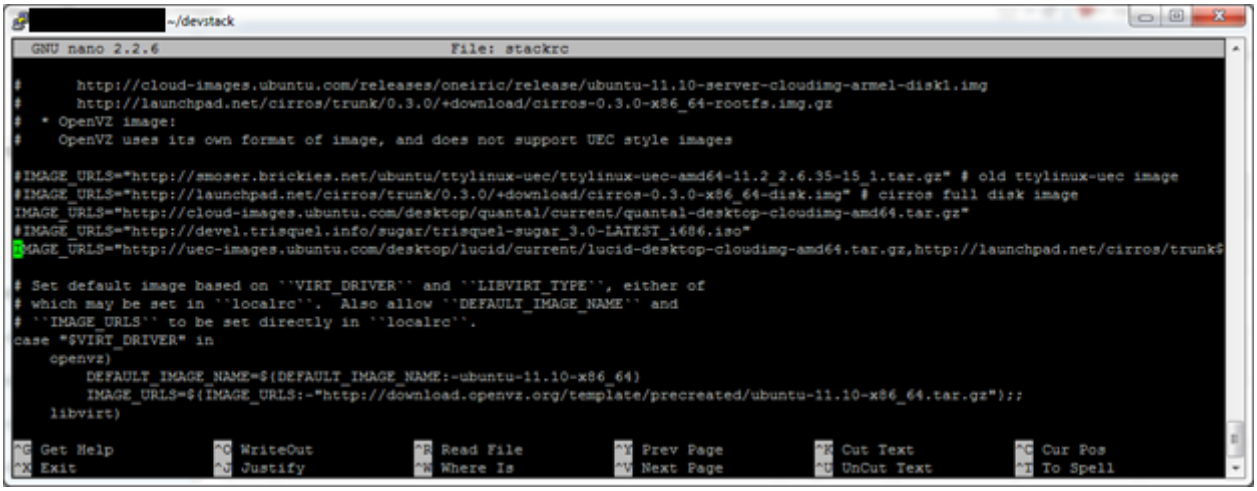


This is the what the text file should look like.



Scroll all the way down, delete the # in front of the image you want to use. You can also add additional images by seperating each image with a comma.

For example:



Afterwards, reboot Openstack, and the image should appear under the Images page in the GUI.